

We Claim:

1. An edit module comprising:
an edit program construction engine,
wherein the edit program construction engine creates
5 an edit program for a packet in response to a disposition
decision for the packet, and wherein the edit program is applied
to modify the packet.

2. The edit module of claim 1 wherein the edit program
10 includes a plurality of instructions, and wherein one or more
instructions determine one or more data bits to be included in
the modified packet.

3. A method of modifying an inbound packet to generate an
15 outbound packet, the method comprising the steps of:
creating an edit program for the inbound packet in
response to a disposition decision for the inbound packet; and
applying the edit program to the inbound packet to
generate the outbound packet.

20 4. The method of modifying an inbound packet of claim 3
wherein the edit program includes a plurality of instructions,
and wherein the step of applying the edit program comprises the
step of determining one or more data bits to be included in the
25 outbound packet.

5. A packet switching controller for processing an
inbound packet, the packet switching controller comprising:
a first engine for constructing an edit program for
30 the inbound packet in response to a disposition decision for the
inbound packet;
a memory for storing the edit program; and

a second engine for executing the edit program to modify the inbound packet to generate an outbound packet.

6. The packet switching controller of claim 5 wherein the edit program includes a plurality of instructions, and one or more instructions determine a plurality of data bits to be included in the outbound packet.

7. The packet switching controller of claim 5 wherein the edit program includes a plurality of instructions, and one or more instructions are for performing at least one operation selected from the group consisting of RECORD, PLAYBACK, COPY, DELETE, INSERT and OVERWRITE operations.

8. The packet switching controller of claim 5 wherein the edit program includes a plurality of instructions that are executed serially.

9. The packet switching controller of claim 5 wherein the second engine includes a packet input buffer for receiving and for temporarily storing the inbound packet.

10. The packet switching controller of claim 9 wherein the inbound packet is stored in the packet input buffer until the edit program has been constructed for the inbound packet.

11. The packet switching controller of claim 5 wherein the second engine includes a playback buffer for storing data from the inbound packet and for playing back at least a portion of the stored data.

12. The packet switching controller of claim 5 wherein the second engine includes a packet output buffer, which is used to modify one or more bits of the inbound packet to generate the outbound packet, and to transmit the outbound packet.

5

13. A method of processing an inbound packet to generate an outbound packet using a real-time constructed edit program, the method comprising the steps of:

constructing the edit program for the inbound packet
10 in response to a disposition decision for the inbound packet;

storing the edit program in a memory;

modifying the inbound packet by executing the edit
program to generate the outbound packet.

14. The method of processing an inbound packet of claim 13
15 wherein the step of modifying the inbound packet includes the
step of determining a plurality of data bits to be included in
the outbound packet.

15. The method of processing an inbound packet of claim 13
20 wherein the step of modifying the inbound packet includes the
step of performing at least one operation selected from the
group consisting of RECORD, PLAYBACK, COPY, DELETE, INSERT and
OVERWRITE operations.

25

16. The method of processing an inbound packet of claim 13
wherein the step of modifying the inbound packet includes the
step of serially executing a plurality of instructions.

17. The method of processing an inbound packet of claim 13
30 further comprising the steps of receiving the inbound packet and
temporarily storing the inbound packet.

18. The method of processing an inbound packet of claim 17 wherein the inbound packet is stored until the edit program has been constructed for the inbound packet.

5

19. The method of processing an inbound packet of claim 13 further comprising the steps of storing data from the inbound packet and playing back at least a portion of the stored data.

10 20. The method of processing an inbound packet of claim 13 wherein the step of modifying the inbound packet includes the step of modifying one or more bits of the inbound packet.

21. A switch comprising a switching backplane and a
15 plurality of packet switching controllers, one or more packet switching controllers comprising:

a buffer for receiving and storing an inbound packet;

a first engine for constructing an edit program real-time using a disposition decision for the inbound packet; and

20 a second engine for executing the edit program to modify the inbound packet into an outbound packet,

wherein the packet switching controller that modifies the inbound packet transmits the outbound packet over the switching backplane to one or more of other packet switching
25 controllers.

22. A switch comprising a switching backplane and a plurality of packet switching controllers, one or more packet switching controllers comprising:

30 means for receiving and storing an inbound packet;

means for constructing an edit program real-time using a disposition decision for the inbound packet; and

means for executing the edit program to modify the inbound packet,

wherein the packet switching controller that modifies the inbound packet transmits the outbound packet over the switching backplane to one or more of other packet switching controllers.

2025 11 11 11:11:11